Anonymity Guarantees of the UMTS/LTE Authentication and Connection Protocol

Ming-Feng Lee, Nigel P. Smart, Bogdan Warinschi, and Gaven J. Watson

University of Bristol

Abstract. The UMTS/LTE protocol for mobile phone networks has been designed to offer a limited form of anonymity for mobile phone uses. In this paper we quantify precisely what this limited form of anonymity actually provides via a formal security model. The model considers an execution where the home and roaming network providers are considered as one entity. We consider two forms of anonymity, one where the mobile stations under attacked are statically selected before the execution, and a second one where the adversary selects these stations adaptively. We prove that the UMTS/LTE protocol meets both of these security definitions. Our analysis requires new assumptions on the underlying keyed functions for UMTS, which whilst probably true have not previously been brought to the fore.

1 Introduction

The Global System for Mobile Communications (GSM) developed by the European Telecommunications Standards Institute (ETSI) was the first cellular communication system designed to provide user authentication and data confidentiality. To overcome the security weaknesses in the original GSM systems, a new standard, called the Universal Mobile Telecommunication System (UMTS), was developed by the Third-Generation Partnership Project (3GPP); this created a system commonly referred to as "3G". In the last few years a further update called UMTS Long Term Evolution (LTE) [18] has also been introduced by 3GPP. LTE is also referred as EUTRA (Evolved UMTS Terrestrial Radio Access) or E-UTRAN (Evolved UMTS Terrestrial Radio Access Network), but usually is commonly called "4G"

The 2G, 3G and 4G systems divide the players in a protocol into various entities; the mobile phone (called Mobile Station in the standards), the home network of the user and the roaming, or serving, network. The serving network is often represented as the base station to which the phone is currently talking. This distinction between different network operators is to enable a phone user to "roam", and thus use their phone in different countries without needing to continually route traffic back to the home network. The basic UMTS/LTE authentication and key agreement protocol (known as AKA) retains the framework of the GSM AKA, but provides enhanced security properties. In particular the AKA protocol aims to provide entity authentication, data confidentiality and data integrity. These properties are provided by the AKA protocol.

An additional security goal was to provide a limited form of anonymity for the user. Each user is identified by a permanent identity, called an IMSI (International Mobile Subscriber Identity). The protocol aims to minimize the use of the IMSI and instead replace the IMSI with a temporary identity, called a TMSI (Temporary Mobile Subscriber Identity). The security goal related to anonymity is to try to stop the IMSI and TMSI's used by a given mobile phone from being linked. If they could be linked a user could then be tracked through the network.

This paper aims to clarify what security properties the 3G/4G protocols provide in terms of this goal of user anonymity. In doing so we provide a formal security model, and prove that the protocol suite satisfies this model. Along the way we provide a precise description of the security properties required of the underlying keyed functions used in the UMTS/LTE protocol, which may be of independent interest.

Prior Security Analysis: A lot of prior analysis of the security of GSM/UMTS/LTE has gone into the properties of the underlying cryptographic functions, [7, 17, 20]. This work is orthogonal to the issues we are interested in. Our focus is on the protocols in which these functions are used, and to derive the required security properties which the functions need to provide so as to guarantee the protocol security properties.

During the change over from 2G to 3G networks there were a number of possible attacks on the protocol. For example [22] present a man-in-the-middle attack, which uses the lack of mutual authentication and integrity protection in GSM, to enable an active adversary to impersonate a legitimate GSM base station and hence forge a cipher mode command message. This allows the adversary to cheat a victim mobile phone into using either no encryption or a weak encryption algorithm such as A5/2 in GSM.

Zhang and Fang [23] pointed out that UMTS AKA is vulnerable to a redirection attack, a variant of a false base station attack. If an adversary owns a device which has the functionality of a base station, the adversary can impersonate as a genuine base station and then map the victim mobile phone on the false base station. As a consequence, the adversary can redirect the outgoing traffic of the victim phone from one network to different one. Furthermore, Zhang and Fang exposed an active attack by a corrupted network in which the adversary can mount a false base station attack to impersonate another uncorrupted network.

We now turn to prior analysis of methods to circumvent the anonymity guarantees. Arapinis et al [6] exposed two such threats. In the first attack, called an IMSI paging attack, the adversary attacks the paging procedure used to locate the phone. If the temporary identity TMSI of the phone is not known by the serving network, the permanent identity IMSI is used to identify the phone. By injecting a paging request multiple times and observing the multiple replies, an active adversary can correlate the paged IMSI and related TMSI of a victim mobile phone in the area covered by adversary's device (false base station). Arapinis et al also provide an analysis, via formal methods, of a modified version of the UMTS protocol and show this meets an notion of anonymity.

In the second attack in [6], called a AKA protocol linkability attack, an active adversary which has previous intercepted an authentication request message can replay the message and check the presence of a specific phone in a particular area. Because the victims mobile phone will return a synchronization failure message after receipt of the replayed authentication request message, the adversary can trace the movements of the victim mobile phone.

Finally, the IMSI catcher attack [19] makes use of the fact that the IMSI of a mobile phone is sent in cleartext when the phone is registering for the first time in the serving network. This kind of attack can lead a mobile phone to reveal its IMSI by triggering the identification procedure from a false base station to the victim mobile phone.

Contributions: As already remarked, anonymity in the UMTS/LTE protocol suite succumbs to a number of attacks, with most attacks relying on the use of a corrupted base station. However, whether it is secure against adversaries which do not corrupt any of the network participants is still worth investigating. To our knowledge no security analysis has been conducted for the anonymity requirement against adversaries who may intercept, transmit and replay messages between phones and the network, but who are not able to impersonate either the roaming or home networks. In addition, most of the previous studies only concentrate on the UMTS/LTE AKA; they fail to consider the security of the whole authentication and connection establishment protocol. The security of data transmission and TMSI allocation (which allocates temporary identities to mobile phones for anonymity) followed by connection establishment are never considered.

In this paper, we focus on anonymity property of the UMTS/LTE at "protocol level" against such adversaries. To formally analyze the protocol, we first give a modified two-party protocol which captures the security properties that UMTS/LTE the authentication and connection protocol provides. Since we

focus on the security on the radio access link and the links between serving network and home environment are assumed be adequately secure. We therefore consider the serving network and home environment as a single party, which we call "the" network. Since a home network can always trace a user (since bills need to be paid) we can restrict to networks which are honest. This assumption eliminates any attack which requires an adversary to successfully impersonate a base station or roaming network. We feel this strong assumption is justified as any adversary which includes a roaming network can break confidentiality of the phone conversation, which is a more important security property than the mild anonymity requirements envisaged by the designers of the protocol.

The anonymity notion we provide includes not only protecting a user's identity but also protects against linking of protocol transactions. The two party protocol we consider not only includes the AKA and connection establishment phases, but also includes phases related to TMSI allocation and data transmission after the authentication and connection establishment. We then propose a security model which captures the anonymity property provided by our two party variant of UMTS/LTE.

Intuitively, anonymity means that a user can identify herself, communicate or use some service without leaking her identity. Up to now, anonymity has been formally defined for various cryptographic schemes in the literature, for example the definition of anonymity for group signatures [9, 11], ring signature [12], ad hoc anonymous identification [16], and direct anonymous communication (DAA) [13, 14].

Typically, one adopts an indistinguishability based formalization to define anonymity. In such a model, the adversary selects two identities $(\mathsf{id}_{i_0}, \mathsf{id}_{i_1})$ to be challenged, then the adversary queries a challenge oracle with a hidden bit $b \in \{0, 1\}$ just once and is returned a signature or public transcript with respect to id_{i_b}. Generally, the target signature or transcript is produced by using the key of the user with id_{i_b}. The goal of the adversary in such anonymity model is trying to determine the hidden bit b. It is required that the adversary has negligible advantage over one-half in distinguishing the two identities from the given signature or transcript.

We also adopt the indistinguishability based formalization for UMTS/LTE protocol but with two slight modifications. In UMTS/LTE protocol, the mobile phone will be allocated a new TMSI after a TMSI *Allocation* procedure and then uses the new TMSI to identify itself when interacting with the network. For privacy, an adversary should not link to the same user two transcripts where one transcript is generated before TMSI *Allocation* and the other is produced after TMSI *Allocation*. To model this kind of interaction, instead of a challenge oracle, our model has a challenge phase in which the adversary is given two just allocated TMSI (TMSI_{i_b} and $\mathsf{TMSI}_{i_{1-b}}$) in random order at the beginning of this phase and can then perform queries to some oracles multiple times with TMSI_{i_b} or $\mathsf{TMSI}_{i_{1-b}}$. Thus our model concerns not only identity confidentiality but also user unlinkability.

In addition our security model bears a close relationship to those used for key agreement, e.g. the BR-style models [8, 10, 15]. We can think of the TMSI as analogous to a secret key and the adversary is trying to determine which session a secret key belongs to. In particular our model has an analogue of the Reveal queries used in key agreement security to enable the adversary to determine TMSI's of sessions on which he is not being challenged.

We further refine the anonymity definition with two subcases. One subcase is the static case, in which the adversary is given two fixed phone identities and then tries to distinguish them from message transmissions. The other is for the dynamic case in which the adversary can dynamically choose two identity indexes of phones on which to be challenged. Our first result shows that if the underlying primitives are secure, then the protocol indeed meets our anonymity requirement for the static case. Our second result shows that if the protocol is anonymous for the static case, then it is also anonymous for the dynamic case.

To end this introduction we overview what we meant above by the underlying primitives to be secure. The UMTS/LTE protocol makes use of a variety of keyed cryptographic functions, commonly referred to as {f1, f2,

$\mathsf{f3}, \mathsf{f4}, \mathsf{f5}, \mathsf{f8}, \mathsf{f9}\}$. These functions are used to generate keys, authenticate messages and provide confidentiality. Informally it would appear that one needs to model the functions as Pseudo Random Functions (PRFs). However, the function subset $\{\mathsf{f1}, \mathsf{f2}, \mathsf{f3}, \mathsf{f4}, \mathsf{f5}\}$ whilst distinct all take the same key as input. Thus our requirement is that this set is "PRF Agile", where we use agile in the sense of Acar et al [5].

2 The UMTS/LTE Protocol Stack

Overview of Protocol Execution: The UMTS/LTE protocol stack contains two main security protocols aimed at authentication and connection establishment. The overall goal is to establish a secure channel between the phone (a.k.a. mobile station (MS)) and "the network". The network is actually a combination of parties consisting of a visitor location register/serving GPRS Support Node (VLR/SGSN) and a serving radio network controller (SRNC) where SRNC is the base station controller of the serving network; but for our purposes we will, as described in the introduction, consider the whole network as a single entity called "the network".

The UMTS/LTE authentication and connection establishment protocol, the communication procedures between the phone and the network can be classified into the following two cases.

- *Case 1. The phone and serving network have already agreed on integrity and cipher keys* IK *and* CK*. The phone also has been assigned a temporary identity* TMSI *by the network:* In this case, the phone first identifies itself by means of the key set identifier KSI and its temporary identity TMSI. The key set identifier KSI enables the network to identify the integrity and cipher keys which are stored in the phone database without invoking the authentication procedure. If the phone and the network need to rekey, then they perform the AKA to agree new keys. Otherwise, the phone and the network use previously agreed keys and algorithms to communicate with each other.
- *Case 2. The phone and the network have not share the common keys integrity and cipher keys. The phone has not been assigned* TMSI *by the serving network:* In this case, the phone needs to identify itself by means of its permanent identity IMSI and then it performs the AKA to share an integrity key IK and cipher key CK with the network. Then the phone and the network then use IK to negotiate the integrity and ciphering algorithms. After the connection set-up, the phone and the network can do secure *Data Transmission* or TMSI *Allocation* to allocate a new temporary identity TMSI (the TMSI is encrypted by means of CK) to the phone from the network.

Here we just focus on *Case 2*, since the analysis of *Case 1* is simpler than that of *Case 2* as *Case 1* does not need to consider the security of the AKA protocol. Note that the allocation of a TMSI means that the phone can identify itself by the TMSI so as to achieve anonymity.

Assume a phone wants to establish a secure connection with the serving network. First of all, a message including various parameters and a START value is sent to the network. The parameters include the precise definitions of the integrity and encryption algorithms supported by the phone. The START value acts like a counter, and it is associated with the integrity and cipher keys that were generated during the last authentication. With a new authentication and key agreement (AKA), the START value is initialize to zero. The network (actually the SRNC) then stores the START values and the list of supported algorithms.

If the phone and the network have executed AKA in a previous connection and the phone has been allocated a TMSI, the phone is then identified via the existing TMSI. If the phone has not authenticated itself previous, the phone needs to identified itself via it's IMSI.

Then the parties run the AKA protocol to achieve authentication and agreement of integrity IK and cipher keys CK with the network. For data integrity and confidentiality, the network decides which encryption and

integrity algorithms that are allowed to be used. The integrity algorithms are based on either Kasumi or SNOW, whilst the encryption algorithms are either none, based on Kasumi or based on SNOW. The network chooses the highest preference integrity and encryption algorithms from the list of allowed algorithms that matches any of the algorithms supported by the phone. The network then initiates integrity and ciphering.

Next, the network generates a random number FRESH and a *Security Mode Command* message with a message authentication code MAC-I to the phone. This message includes the security capability of the mobile equipment, the GSM ciphering capability, the selected integrity algorithm and the random number FRESH to be used. If ciphering shall be started, this message also contains the selected encryption to be used. The message authentication code MAC-I is generated by using the integrity key IK and the selected integrity algorithm.

On receiving this command the phone verifies the validity of the received message and MAC-I by means of IK and the indicated integrity algorithm. If the verification passes, the phone generates a *Security Mode Complete* message and a message authentication code MAC-I for this message. Then the phone sends the *Security Mode Complete* message with the message authentication code to the network. If the control of security capability is not successful, the phone ends the procedure.

At reception of the *Security Mode Complete* message, the network verifies the validity of the the received message authentication code MAC-I by using integrity key IK and the indicated integrity algorithm.

Ciphering and Integrity Methods: The ciphering and integrity methods are provided by two functions, f8 and f9 respectively. The use of block cipher Kasumi for f8 and f9 is specified in ETSI TS 35.201 [1] and ETSI TS 35.202 [2], whilst the use of the stream cipher SNOW 3G is specified in ETSI TS 35.215 [3] and ETSI TS 35.216 [4]. For further details see [18].

To encrypt a message, the phone or the network first computes a keystream $\mathsf{KEYSTREAM} = \mathsf{f8}_{\mathsf{CK}}(\mathsf{COUNTER\text{-}C}, \mathsf{BEARER}, \mathsf{DIRECTION}, \mathsf{LENGTH})$, where CK is the cipher key, COUNTER-C is a time-depended counter, BEARER is the radio bearer identifier (the radio bearer identifier BEARER is 5 bits long and will not concern us), DIRECTION is a transmission direction bit, LENGTH is a 16 bit indicator that determines the length of keystream block. After the keystream is computed, the ciphertext is calculated as $\mathsf{CIPHERTEXT} = \mathsf{KEYSTREAM} \oplus \mathsf{PLAINTEXT}$. To decrypt a ciphertext, the phone or the network first computes a keystream and then derives the plaintext $\mathsf{PLAINTEXT} = \mathsf{KEYSTREAM} \oplus \mathsf{CIPHERTEXT}$.

To achieve integrity, a message authentication code is attached with the message to be integrity protected. The message authentication code is computed as

$$\mathsf{MAC\text{-}I} = \mathsf{f9}_{\mathsf{IK}}(\mathsf{COUNTER\text{-}I}, \mathsf{MESSAGE}, \mathsf{DIRECTION}, \mathsf{FRESH}),$$

IK is the integrity key, COUNTER-I is an integrity sequence number, DIRECTION is a direction bit, FRESH is a random value,

In both cases the direction bit DIRECTION is set to zero in the case of a message from the phone to the network, and set to one for the other direction.

The UMTS/LTE AKA Protocol: We now describe in more detail the AKA protocol and the parameters used inside it. Each SIM card in the phone and the authentication center in the user's home environment share a long-term secret key K. Two counters, MS.SQN and NET.SQN are also maintained by the phone and the network respectively to support network authentication. The sequence number NET.SQN is an individual counter for each user and the counter MS.SQN is the highest sequence number the phone has accepted. The initial values for MS.SQN and NET.SQN are set to zero, with the two counters incrementing during each

authentication. Intuitively the two sequence numbers MS.SQN and NET.SQN are used to guarantee the freshness of the AKA protocol.

The value START sent by the phone is used to generate counters COUNTER-I and COUNTER-C in integrity and ciphering algorithms. When the counters COUNTER-I and COUNTER-C are generated, the value START is also updated according to these counters. For more details about the generation and updating of START, COUNTER-I and COUNTER-C, please refer to [18].

The AKA protocols makes use of a set of three message authentication functions $\{\mathsf{f1}, \mathsf{f1}^*, \mathsf{f2}\}$, and four key generation functions $\{\mathsf{f3}, \mathsf{f4}, \mathsf{f5}, \mathsf{f5}^*\}$, all controlled by the use of the same key. In what follows we will not concern ourselves with $\mathsf{f1}^*$ and $\mathsf{f5}^*$, as they are simply variants of f1 and f5 when used in the case of resynchronization; hence we will assume them identical to f1 and f5 in our analysis.

The protocols consist of two subprocedures: The first is for the distribution of authentication data from user's home environment to the serving network and the second is for authentication and key agreement.

The distribution of the data from the home to serving network is not of interest to us, since we subsume the home and serving network into one entity in our security model. However, the output of this procedure is vital to the understanding of what follows. The serving network obtains an ordered array of fresh authentication vectors $\mathsf{AV}(1 \ldots n)$; each authentication vector (called a quintet in UMTS/LTE standard). The authentication vectors are ordered based on sequence numbers. The reason for the serving network obtaining an array of such vectors is to enable it to perform multiple authentications with the phone, without needing to recontact the home network. In our simplification of assuming a single network provider we can assume that fresh individual authentication vectors are obtained for each invocation as opposed to an array of authentication vectors.

The authentication vectors AV are produced as follows: The network starts with generating an unpredictable random number RAND and a fresh sequence number SQN which is derived from NET.SQN. Typically, the sequence number SQN consists of two concatenated parts $\mathsf{SQN} = \mathsf{SEQ}\|\mathsf{IND}$. The index value IND is incremented by one for each new authentication vector to be generated and may range from zero to a where a is the size of the array, and so will not concern us. The generation of SEQ depends on different options provided by the ETSI specification, see [18] for details. In our analysis we abstract this away into an algorithm SQN.Gen which takes as input NET.SQN. The network computes a message authentication code MAC, an expected response XRES, a cipher key CK, and integrity key IK as follows.

- Message authentication code $\mathsf{MAC} = \mathsf{f1}_\mathsf{K}(\mathsf{SQN}\|\mathsf{RAND}\|\mathsf{AMF})$. The AMF field defines operator-specific options in the authentication process, e.g., the lifetime of integrity and cipher keys.
- Expected response $\mathsf{XRES} = \mathsf{f2}_\mathsf{K}(\mathsf{RAND})$.
- Cipher key $\mathsf{CK} = \mathsf{f3}_\mathsf{K}(\mathsf{RAND})$.
- Integrity key $\mathsf{IK} = \mathsf{f4}_\mathsf{K}(\mathsf{RAND})$.
- Anonymity key $\mathsf{AK} = \mathsf{f5}_\mathsf{K}(\mathsf{RAND})$ or $\mathsf{AK} = 0$. (the anonymity key AK is used to conceal the sequence number as the latter may expose the identity and location of the phone, if no concealment of the sequence number is needed then $\mathsf{AK} = 0$).

The network then assembles an authentication token

$$\mathsf{AUTN} = \mathsf{SQN} \oplus \mathsf{AK}\|\mathsf{AMF}\|\mathsf{MAC}$$

and an authentication vector $\mathsf{AV} = \mathsf{RAND}\|\mathsf{XRES}\|\mathsf{CK}\|\mathsf{IK}\|\mathsf{AUTN}$.

The second subprocedure consumes the authentication vectors. The serving network first selects a fresh authentication vector AV. It then sends the random challenge RAND and the authentication token AUTN

from the selected authentication vector AV to the phone. Upon receipt of RAND and AUTN, the phone first computes the anonymity key $\mathsf{AK} = \mathsf{f5}_{\mathsf{K}}(\mathsf{RAND})$ and retrieves the sequence number $\mathsf{SQN} = (\mathsf{SQN} \oplus \mathsf{AK}) \oplus \mathsf{AK}$ from the authentication token $\mathsf{AUTN}(= \mathsf{SQN} \oplus \mathsf{AK} \| \mathsf{AMF} \| \mathsf{MAC})$. Next, the phone computes $\mathsf{XMAC} = \mathsf{f1}_{\mathsf{K}}(\mathsf{SQN} \| \mathsf{RAND} \| \mathsf{AMF})$ and compares this with MAC which is included in the received AUTN. If they are different, the user ends the procedure. The phone also verifies whether the received sequence number SQN is in the correct range, for example, $\mathsf{SQN} > \mathsf{MS.SQN}$. If the phone considers the sequence number SQN is not in correct range, it abandons the procedure. If the sequence number is considered to be in the correct range, the phone computes a response $\mathsf{RES} = \mathsf{f2}_{\mathsf{K}}(\mathsf{RAND})$ and includes it in a *User Authentication Response* message back to the network. Finally, the phone computes a cipher key $\mathsf{CK} = \mathsf{f3}_{\mathsf{K}}(\mathsf{RAND})$ and a integrity key $\mathsf{IK} = \mathsf{f4}_{\mathsf{K}}(\mathsf{RAND})$.

Upon receipt of the RES, the serving network compares RES with the expected response XRES from the authentication vector AV. If RES is the same as XRES, the phone passes the authentication. The serving network then extracts the cipher key CK and integrity key IK from the selected authentication vector. If RES and XRES are different, network abandons the authentication procedure.

3 Security model

In this section we present our security model. We first describe the basic notation, then go onto describe the various oracles which model how the adversary can interact with the UMTS/LTE protocol. We then discuss how these oracles are used to define our security experiments in the two cases of static and dynamic adversaries.

Basic Notation: If S is a set, we denote the act of sampling from S uniformly at random and assigning the result to the variable x by $x \xleftarrow{\$} S$. We let $\{0,1\}^t$ denote the set of binary strings of length t. If A is an algorithm, we write $x \leftarrow A(y_1, \ldots, y_n)$ to indicate that x is obtained by invoking A on inputs $y_1, \ldots, y_n$. The algorithms that we consider may have access to some oracles. We write $\mathcal{A}^{\mathcal{O}}$ to indicate that the algorithm $\mathcal{A}$ has access to oracle $\mathcal{O}$. We also denote concatenation of two date strings x and y as $x \| y$.

Assume $\mathcal{U} = \{\mathsf{MS}_1, \ldots, \mathsf{MS}_m\}$ is the set of all phones (a.k.a. mobile stations) that register to the network. We define IMSI_i to be the permanent identity of MS_i and TMSI_i the temporary identity of MS_i. Let **ID** be a m dimensional array which is initially set to hold $\mathbf{ID}_i = \mathsf{IMSI}_i$, as the protocol proceeds this will be updated to TMSI_i if a phone has been allocated this temporary identity. We also let **Revealed** denote an m dimensional vector of boolean values; which are initially all set to be false. We also let **K**, **MS.SQN**, **NET.SQN**, **START** denote vectors of length m, where **K** is the vector of all master keys, **MS.SQN** is the vector of phone sequence numbers, **NET.SQN** the vector of sequence numbers which the network keeps for all phones, and **START** the vector of all start values. For example with index i, $\mathbf{K}_i = \mathsf{K}_i$ is the master key of the MS_i (the phone with $\mathsf{IMSI}_i/\mathsf{TMSI}_i$).

The following algorithms are used in the oracles and experiments, to abstract away various generation algorithms whose details do not concern us, but whose outputs are needed to define various quantities. The precise definitions of these algorithms can be found in the UMTS/LTE standards.

- Setup(): for every $\mathsf{MS}_i \in \mathcal{U}$, the setup algorithm that generates master keys K_i, initial sequence numbers ($\mathsf{MS.SQN}_i$, $\mathsf{NET.SQN}_i$) and initial START_i values for the phone.
- SQN.Gen: the algorithm that takes as input NET.SQN and output SQN.
- FRESH.Gen: the algorithm that generates a fresh number FRESH.
- COUNTER-I.Gen: the algorithm that takes as input START and output counter COUNTER-I which is used in integrity algorithm.

– COUNTER-C.Gen: the algorithm that takes as input START and output counter COUNTER-C which is used in encryption algorithm.
– START.Update: the algorithm that takes as input the values (START, COUNTER-I/COUNTER-C) and takes as output an updated value for START.

Adversarial oracles: In our security analysis, there are two adversarial oracles which model the behaviour of the phone (the MS oracle) and the network (the NET oracle), and one which allows the adversary to obtain the current identity (either the IMSI or TMSI) of a specific mobile (the Reveal oracle). Both the NET and MS oracle contain program counters for each phone MS_i. The m-vector **NET.pc** is the vector of all program counters that the NET oracle maintains. The element $\mathbf{NET.pc}_i = \mathsf{NET.pc}_i$ denoting the program counter associated to mobile station i. Similarly **MS.pc** is the vector of all program counters that the MS oracle maintains.

We assume there are two globaly defined identities i_0 and i_1, which informally indicate on which phones the adversary is being challenged. How these are set will be defined later when we discuss the security experiments. At this point note that $i_0, i_1 \in \{1, \ldots, m, \perp\}$. We let $\Upsilon = \{\mathsf{f1}, \mathsf{f2}, \mathsf{f3}, \mathsf{f4}, \mathsf{f5}\}$ be the set of functions with the same key used in UMTS/LTE and define $\mathsf{R} = \{\Upsilon, \mathsf{f8}, \mathsf{f9}\}$.

The mobile station and net oracles are defined as in Figure 1. The oracles $\mathsf{NET}[\mathsf{X},\mathsf{Y}](\mathsf{id}, x)$, oracle $\mathsf{MS}[\mathsf{X},\mathsf{Y}](\mathsf{id}, x)$ are parametrized by two sets X and Y, as well as taking in two inputs id and x. The oracles $\mathsf{NET}[\mathsf{X},\mathsf{Y}](\mathsf{id}, x)$ and $\mathsf{MS}[\mathsf{X},\mathsf{Y}](\mathsf{id}, x)$ run the functions $\mathsf{net}[\mathsf{X}](\mathsf{K}_i, \mathsf{NET.pc}_i, x)$ and $\mathsf{ms}[\mathsf{X}](\mathsf{K}_i, \mathsf{MS.pc}_i, x)$ respectively when the index of identity id is i_0. If the index of identity id is i_1, then the $\mathsf{NET}[\mathsf{X},\mathsf{Y}](\mathsf{id}, x)$ and $\mathsf{MS}[\mathsf{X},\mathsf{Y}](\mathsf{id}, x)$ oracles run the functions $\mathsf{net}[\mathsf{Y}](\mathsf{K}_i, \mathsf{NET.pc}_i, x)$ and $\mathsf{ms}[\mathsf{Y}](\mathsf{K}_i, \mathsf{MS.pc}_i, x)$ respectively. If the index of identity id is neither i_0 nor i_1, then the NET and MS oracles run the functions $\mathsf{net}[\mathsf{R}](\mathsf{K}_i, \mathsf{NET.pc}_i, x)$ and $\mathsf{ms}[\mathsf{R}](\mathsf{K}_i, \mathsf{MS.pc}_i, x)$ respectively.

Oracle $\mathsf{NET}[\mathsf{X},\mathsf{Y}](\mathsf{id}, x)$
– find i such that $\mathsf{id} = \mathbf{ID}_i$, otherwise abort
– if $i = i_0$, $y \leftarrow \mathsf{net}[\mathsf{X}](\mathsf{K}_i, \mathsf{NET.pc}_i, x)$
– else $i = i_1$, $y \leftarrow \mathsf{net}[\mathsf{Y}](\mathsf{K}_i, \mathsf{NET.pc}_i, x)$
– else $y \leftarrow \mathsf{net}[\mathsf{R}](\mathsf{K}_i, \mathsf{NET.pc}_i, x)$
– return y

Oracle $\mathsf{MS}[\mathsf{X},\mathsf{Y}](\mathsf{id}, x)$
– find i such that $\mathsf{id} = \mathbf{ID}_i$, otherwise abort
– if $i = i_0$, $y \leftarrow \mathsf{ms}[\mathsf{X}](\mathsf{K}_i, \mathsf{MS.pc}_i, x)$
– else $i = i_1$, $y \leftarrow \mathsf{ms}[\mathsf{Y}](\mathsf{K}_i, \mathsf{MS.pc}_i, x)$
– else $y \leftarrow \mathsf{ms}[\mathsf{R}](\mathsf{K}_i, \mathsf{MS.pc}_i, x)$
– return y

Fig. 1. NET and MS oracles defining security for modified UMTS/LTE authentication and connection protocol

In the real world the sets X and Y will also be equal to R. However, we generalise the notation to allow X and Y to be different so as to provide a notational simplification for our security proof which follows. The functions net and ms used by the oracles are given in Figures 5 and 6 of Appendix B. We present a textual overview here, but the reader should simply think of the oracles/functions as implementing the UMTS/LTE protocol definition but for abstract function sets $\{\{\mathsf{h1}, \mathsf{h2}, \mathsf{h3}, \mathsf{h4}, \mathsf{h5}\}, \mathsf{h8}, \mathsf{h9}\}$, which may or may not be equivalent to the functions used in the real protocol.

To be able to call the NET and MS oracles for the ith phone the adversary needs access to the current value of $\mathbf{ID}_i$. This is done via means of the Reveal oracle, on input of an index $i \in \{1, \ldots, m\}$ this returns the current value of $\mathbf{ID}_i$ and sets $\mathbf{Revealed}_i$ to be true.

The oracle NET gives the adversary the ability to communicate with the network. The adversary can send message x to the network, claiming x is from the phone with id by calling the oracle on input (id, x). The oracle maintains **ID**, **K**, **NET.SQN**, **START**, **COUNTER-I**, **COUNTER-C** and **NET.pc**. The oracle

also uses program counter $\mathsf{NET.pc}_i$ to maintain the state of the oracle for each phone. If $\mathsf{NET.pc}_i = 1$, the NET oracle receives the security capability of the phone (supported integrity and cipher algorithms of the phone) from the phone and then starts user authentication. If $\mathsf{NET.pc}_i = 2$, the oracle receives *User Authentication Response*. If $\mathsf{NET.pc}_i = 3$, the oracle receives *Security Mode Complete*. If $\mathsf{NET.pc}_i = 4$, the oracle starts TMSI *Allocation*. If $\mathsf{NET.pc}_i = 5$, the oracle receives TMSI *Allocation Complete*. If $\mathsf{NET.pc}_i = 6$, the oracle starts *Data Transmission*. If $\mathsf{NET.pc}_i = 7$, the oracle receives transmitted data. Note after AKA and the negotiation of integrity and encryption algorithms have been finished, the phone and the network can run TMSI *Allocation* or *Data Transmission*. In order to express the branch of the adversary's decisions, after receiving *Security Mode Complete*, we ask the adversary designate the next state (TMSI *Allocation Command*/start *Data Transmission*/receive transmitted data) he wants the oracle to go by appending an additament pc with x. This additament pc effects the program counter $\mathsf{NET.pc}_i$ and decides next oracle state.

The oracle MS gives the adversary the ability to communicate with the phone. The adversary can send message x to the phone with id by calling the oracle on input (id, x). The oracles maintains **ID**, **K**, **MS.SQN**, **START**, **COUNTER-I**, **COUNTER-C** and **MS.pc**. The oracle also uses program counter $\mathsf{MS.pc}_i$ to maintain the state of the oracle for each phone. If $\mathsf{MS.pc}_i = 1$, the MS oracle starts communication. If $\mathsf{MS.pc}_i = 2$, the oracle receives *User Authentication Request* and *User Authentication Response*. If $\mathsf{MS.pc}_i = 3$, the oracle receives *Security Mode Command*. If $\mathsf{MS.pc}_i = 4$, the oracle receives TMSI *Allocation Command*. If $\mathsf{MS.pc}_i = 5$, the oracle starts *Data Transmission*. If $\mathsf{MS.pc}_i = 6$, the oracle receives transmitted data. we also ask the adversary designate the next state (TMSI *Allocation Complete*/start *Data transmission*/receive transmitted data) he wants the oracle to go by appending an additament pc with x. This additament pc effects the program counter $\mathsf{MS.pc}_i$ and decides next oracle state.

Security Experiments: It is clear that the authentication and connection protocol does not offer any form of strong anonymity; indeed it is not designed to. For example, when a phone communicates with the serving network at the first time, the phone needs to identify itself by its permanent identity IMSI and the downlink or uplink message is sent with a tag of IMSI. Therefore, anonymity does not hold in the first communication between the phone and the network. In addition during the TMSI *Allocation/Reallocation* procedure, the network sends a TMSI *Allocation Command* containing the encrypted new temporary identity TMSI_n, the phone then returns a TMSI *Allocate Complete* acknowledgment. If the network does not receive the TMSI *Allocation Complete* acknowledgment from the phone, the network falls back to using IMSI for downlink signaling and the phone should identify itself by its permanent identity IMSI again.

However, outside of these two cases and in the case of an honest network UMTS/LTE should offer anonymity and unlinkability of communications. We first consider the case of an adversary $\mathcal{A}$ that controls the communication between the network and two fixed phones MS_{i_0} and MS_{i_1}. The adversary can eavesdrop on transcripts or send its own messages to get responses from the phones and the network. For example, by querying the oracles $\mathsf{MS}[\mathsf{R},\mathsf{R}](\mathsf{id}, x)$ and $\mathsf{NET}[\mathsf{R},\mathsf{R}](\mathsf{id}, x)$ with id corresponds to $\mathbf{ID}_{i_0}$ or $\mathbf{ID}_{i_1}$, the adversary gets backs public transcripts between the network and MS_{i_0} or MS_{i_1}.

We can thus define two kinds of adversaries: passive adversaries and active adversaries. A passive adversary processes as follows: the passive adversary first calls MS oracle with $(\mathsf{IMSI}_i, \mathsf{init})$ to initiate a new communication. If $\mathcal{A}$ sends (id, x) to MS/NET oracle, x should be the output of the former query to NET/MS oracle except when $x = \mathsf{init/allocate/send}$, i.e., the adversary just forwards the messages which MS/NET outputs to NET/MS oracles faithfully until the end of connection establishment. An active adversary can query MS and NET oracles too. However, an active adversary may not simply forward the message but can send arbitrary message to MS/NET oracle in particular. Our security results are in the more stringent case of an active adversary.

Let Υ and R be as before, we then formally, we define an experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}]$ that depends on protocol Π and adversary $\mathcal{A}$. This experiment is used to model the case of static security, where the adversary is told which two phones he will end up attacking. The details are given in Figure 2. The experiment starts by running $\mathsf{Setup}()$ which generates the various required parameters. The experiment procedures in two phases: In the first phase the adversary calls the oracles $\mathsf{MS}[\mathsf{R},\mathsf{R}](\mathsf{id},x)$ and $\mathsf{NET}[\mathsf{R},\mathsf{R}](\mathsf{id},x)$ just as it would in the real world. At the end of this phase, the adversary outputs some state information st. The restriction is that in the end of the first phase, both phones MS_{i_0} and MS_{i_1} must be in an unrevealed state, i.e. $\mathbf{Revealed}_{i_0} = \mathbf{Revealed}_{i_1} = \mathsf{false}$.

The second phase is the challenge phase. In the beginning of this phase, the adversary is given the two just allocated TMSIs $(\mathsf{TMSI}_{i_b}, \mathsf{TMSI}_{i_{1-b}})$ (where $b \in \{0,1\}$) for the two phones $(\mathsf{MS}_{i_0}, \mathsf{MS}_{i_1})$ but in a random order, i.e., $\mathcal{A}$ does not know which TMSI belongs to which phone. Then the adversary can query $\mathsf{MS}[\mathsf{R},\mathsf{R}](\mathsf{id},x)$ and $\mathsf{NET}[\mathsf{R},\mathsf{R}](\mathsf{id},x)$ oracles with (id,x) where $\mathsf{id} = \mathsf{TMSI}_{i_b}$ or $\mathsf{id} = \mathsf{TMSI}_{i_{1-b}}$. In this challenge phase the adversary is not allowed to query the Reveal oracle on the indexes i_0 or i_1, as that would allow him to trivially win the game. At the end of the challenge phase, the adversary outputs a bit $\hat{b}$. This bit $\hat{b}$ is the output of the experiment.

Definition 1 (Anonymity for static case). *Let $\Upsilon = \{\mathsf{f1},\mathsf{f2},\mathsf{f3},\mathsf{f4},\mathsf{f5}\}$ be a set of keyed function with the same secret key,* $\mathsf{f8}$ *and* $\mathsf{f9}$ *be keyed functions, and $\mathsf{R} = \{\Upsilon, \mathsf{f8}, \mathsf{f9}\}$. We define the advantage of an adversary in breaking the anonymity in the static case to be*

$$\mathsf{Adv}^{\text{s-anon}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}] = \left|\Pr[\mathsf{Exp}^{\text{s-anon-0}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}] = 1] - \Pr[\mathsf{Exp}^{\text{s-anon-1}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}] = 1]\right|.$$

$\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}]$
- $(\mathbf{K}, \mathbf{MS.SQN}, \mathbf{NET.SQN}, \mathbf{START}) \leftarrow \mathsf{Setup}()$
- $\mathsf{st} \leftarrow \mathcal{A}^{\mathsf{MS}[\mathsf{R},\mathsf{R}](\mathsf{id},x),\mathsf{NET}[\mathsf{R},\mathsf{R}](\mathsf{id},x),\mathsf{Reveal}(i)}()$
- $\hat{b} \leftarrow \mathcal{A}^{\mathsf{MS}[\mathsf{R},\mathsf{R}](\mathsf{id},x),\mathsf{NET}[\mathsf{R},\mathsf{R}](\mathsf{id},x),\mathsf{Reveal}(i)}(\mathsf{st}, \mathsf{TMSI}_{i_b}, \mathsf{TMSI}_{i_{1-b}})$
- output $\hat{b}$

$\mathsf{Exp}^{\text{d-anon-}b}_{\Pi,\mathcal{A}}[\mathsf{R},\mathsf{R}]$
- $i_0, i_1 \leftarrow \perp$
- $(\mathbf{K}, \mathbf{MS.SQN}, \mathbf{NET.SQN}, \mathbf{START}) \leftarrow \mathsf{Setup}()$
- $(\mathsf{st}, i_0, i_1) \leftarrow \mathcal{A}^{\mathsf{MS}[\mathsf{R},\mathsf{R}](\mathsf{id},x),\mathsf{NET}[\mathsf{R},\mathsf{R}](\mathsf{id},x),\mathsf{Reveal}(i)}()$
- $\hat{b} \leftarrow \mathcal{A}^{\mathsf{MS}[\mathsf{R},\mathsf{R}](\mathsf{id},x),\mathsf{NET}[\mathsf{R},\mathsf{R}](\mathsf{id},x),\mathsf{Reveal}(i)}(\mathsf{st}, \mathsf{TMSI}_{i_b}, \mathsf{TMSI}_{i_{1-b}})$
- output $\hat{b}$

Fig. 2. Experiments defining anonymity for static and dynamic case of UMTS/LTE protocol.

Security in the dynamic case follows a similar model, but now the adversary determines which phones it wants to be challenged on, returning this information to the challenger at the end of the first phase. The two phones have the same restriction as to their state as in the static case. We can formally define an experiment, $\mathsf{Exp}^{\text{d-anon-}b}_{\Pi,\mathcal{A}}[\mathsf{R},\mathsf{R}]$, that depends on protocol Π and adversary $\mathcal{A}$ as in Figure 2. We define the advantage in a similar way for this case via

$$\mathsf{Adv}^{\text{d-anon}}_{\Pi,\mathcal{A}}[\mathsf{R},\mathsf{R}] = \left|\Pr[\mathsf{Exp}^{\text{d-anon-0}}_{\Pi,\mathcal{A}}[\mathsf{R},\mathsf{R}] = 1] - \Pr[\mathsf{Exp}^{\text{d-anon-1}}_{\Pi,\mathcal{A}}[\mathsf{R},\mathsf{R}] = 1]\right|.$$

4 Security Analysis

Our anonymity theorems are conditional in that they depend on the underlying functions $\{\mathsf{f1},\mathsf{f2},\mathsf{f3},\mathsf{f4},\mathsf{f5},\mathsf{f8},\mathsf{f9}\}$ having certain properties. As remarked in the introduction the precise property is complicated by the fact that the functions $\{\mathsf{f1},\mathsf{f2},\mathsf{f3},\mathsf{f4},\mathsf{f5}\}$ are called using the same key. In both definition agility and defining the security requirement for $\mathsf{f8}$ and $\mathsf{f9}$ we will need the notion of a PRF, which we recall next.

Definition 2 (Pseudo-random functions). *Let ℓ_1 and ℓ_2 be positive integers. Let $\mathcal{F} := \{F_s\}$ be a family of keyed functions under key s, where each function F_s maps $\{0,1\}^{l_1}$ to $\{0,1\}^{l_2}$. Let Γ_{ℓ_1,ℓ_2} denote the set of all functions form $\{0,1\}^{l_1}$ to $\{0,1\}^{l_2}$. Consider an adversary $\mathcal{A}$ that has oracle access to a function in Γ_{ℓ_1,ℓ_2}, and suppose that $\mathcal{A}$ always outputs a bit. We define the PRF-advantage of $\mathcal{A}$ to be*

$$\mathsf{Adv}^{\mathsf{prf}}_{F,\mathcal{A}} = |\Pr[F_s \xleftarrow{\$} \mathcal{F} : \mathcal{A}^{F_s}() = 1] - \Pr[f \xleftarrow{\$} \Gamma_{\ell_1,\ell_2} : \mathcal{A}^{f}() = 1]|.$$

The notion of agility [5] considers a set of schemes, all meeting some base notion of security, and requires that security is maintained when multiple schemes use the same key. Agility is thus is not a property of an individual scheme but of a set of schemes relative to some (standard) security notion for these schemes. In our context we take a set Υ of PRFs and talk of their agility with respect to the PRF notion:

Definition 3 (PRF Agility). *Let $\mathcal{F} := \{F\}$ be a family of keyed functions. Let Γ denote the set of all functions. Let $\Upsilon = \{\mathsf{f1}, \mathsf{f2}, ..., \mathsf{f}n\}$ be a subset of $\mathcal{F}$, $\Psi = \{\mathsf{r1}, \mathsf{r2}, \mathsf{r3}, ..., \mathsf{r}n\}$ be a subset of Γ, $|\Upsilon| = |\Psi|$ and the range of $\mathsf{f}j$ is equal to that of $\mathsf{r}j$ $(1 \leq j \leq n)$. Consider an adversary $\mathcal{A}$ that has oracle access to a set of functions in Γ, and suppose that $\mathcal{A}$ always outputs a bit. Define the advantage of $\mathcal{A}$ to be*

$$\mathsf{Adv}^{\mathsf{PR}}_{\Upsilon,\mathcal{A}} = |\Pr[\Upsilon \xleftarrow{\$} \mathcal{F} : \mathcal{A}^{\Upsilon}() = 1] - \Pr[\Psi \xleftarrow{\$} \Gamma : \mathcal{A}^{\Psi}() = 1]|.$$

Anonymity of the protocol for the static and the dynamic case is formalized by the following two theorems; the proofs are in Appendix A.

Theorem 1. *If there is an adversary $\mathcal{A}$ against the static anonymity of the UMTS protocol, then there are adversaries $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$ and $\mathcal{G}$ such that*

$$\mathsf{Adv}^{\mathsf{s\text{-}anon}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}] \leq \mathsf{Adv}^{\mathsf{PR}}_{\Upsilon,\mathcal{B}} + \mathsf{Adv}^{\mathsf{PR}}_{\Upsilon,\mathcal{C}} + \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f8},\mathcal{D}} + \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f9},\mathcal{E}} + \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f8},\mathcal{F}} + \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f9},\mathcal{G}}.$$

Theorem 2. *If the UMTS/LTE authentication and connection protocol is run with a maximum of m phones with an adversary $\mathcal{A}$, then there is an adversary $\mathcal{B}$ which satisfies*

$$\mathsf{Adv}^{\mathsf{d\text{-}anon}}_{\Pi,\mathcal{A}}[\mathsf{R},\mathsf{R}] \leq m(m-1) \cdot \mathsf{Adv}^{\mathsf{s\text{-}anon}}_{\Pi,\mathcal{B},i_0,i_1}[\mathsf{R},\mathsf{R}].$$

5 Acknowledgements

This work has been supported in part by ERC Advanced Grant ERC-2010-AdG-267188-CRIPTO, the second author was also supported in part by a Royal Society Wolfson Merit Award.

References

1. 3GPP TS 35.201. 3rd Generation Partnership Project Technical Specification Group Services and System Aspects; Specification of the 3GPP confidentiality and integrity algorithms; Document 1: f8 and f9 specifications.
2. 3GPP TS 35.202. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Specification of the 3GPP confidentiality and integrity algorithms; Document 2: Kasumi algorithm specification.
3. 3GPP TS 35.215. Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 and UIA2; Document 1: UEA2 and UIA2 specifications.

4. 3GPP TS 35.216. Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 and UIA2; Document 2: SNOW 3G specification.
5. T. Acar, M. Belenkiy, M. Bellare, and D. Cash. Cryptographic Agility and its Relation to Circular Encryption. *Advances in Cryptology — EUROCRYPT '10*, Springer-Verlag LNCS 6110, 403–422, 2010.
6. M. Arapinis, L. Mancini, E. Ritter, M. Ryan, N. Golde, K. Redon and R. Borgaonkar. New Privacy Issue in Mobile Telephony: Fix and Verification Computer and Communications Security – CCS'12, ACM, 205–216, 2012.
7. E. Barkan, E. Biham and N. Keller. Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication. *Advances in Cryptology — CRYPTO '03*, Springer-Verlag LNCS 2729, 600–616, 2003.
8. M. Bellare, R. Canetti and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. *30th Symposium on Theory of Computing – STOC 1998*, ACM, 419–428, 1998.
9. M. Bellare, D. Micciancio and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. *Advances in Cryptology – EUROCRYPT 2003*, Springer LNCS 2656, 614–629, 2003.
10. M. Bellare and P. Rogaway. Entity authentication and key distribution. *Advances in Cryptology – CRYPTO '93*, Springer-Verlag LNCS 773, 232–249, 1994.
11. M. Bellare, H. Shi and C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. *Topics in Cryptology – CT-RSA 2005*, Springer-Verlag LNCS 3376, 136–153, 2005.
12. A. Bender, J. Katz and R. Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. *Theory of Cryptography – TCC 2006*, Springer-Verlag LNCS 3876, 60–79, 2006.
13. E. Brickell, J. Camenisch and L. Chen. Direct anonymous attestation. *Computer and Communications Security – CCS 2004*, ACM Press, 132–145, 2004.
14. E. Brickell, L. Chen and J. Li. Simplified security notions for direct anonymous attestation and a concrete scheme from pairings. *Int. Journal of Information Security*, **8**, 315–330, 2009.
15. R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. *Advances in Cryptology – EUROCRYPT 2001*, Springer-Verlag LNCS 2045, 453–474, 2001.
16. Y. Dodis, A. Kiayias, A. Nicolosi and V. Shoup. Anonymous Identification in Ad Hoc Groups. *Advances in Cryptology – EUROCRYPT 2004*, Springer-Verlag LNCS 3027, 609–626, 2004.
17. O. Dunkelman, N. Keller and A. Shamir. A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony. *Advances in Cryptology – CRYPTO 2010*, Springer LNCS 6223, 393–410, 2010.
18. ETSI TS 133.102 V10.00(2011-05). Universal Mobile Telcommunications System (UMTS); LTE; 3G security; Security architecture (3GPP TS 33.102 10.0.0 Release 10).
19. D. Fox. Der IMSI-Catcher. *Datenschutz und Datensicherheit*, **26(4)**, 212–215, 2002.
20. A. Kircanski. On the sliding property of SNOW 3 G and SNOW 2.0. *IET Information Security*, **5(4)**, 199–206, 2011.
21. C.J. Mitchell. The Security of the GSM Air Interface Protocol. *Technical Report RHUL-MA-2001-3*, Royal Holloway University of London, 2001.
22. U. Meyer and S. Wetzel. A Man-in-the-Middle Attack on GSM. *Workshop on Wireless security – WiSe '04*, ACM, 90–97, 2004.
23. M. Zhang and Y. Fang. Security Analysis and Enhancements of 3GPP Authentication and Key Agreement Protocol. *IEEE Transactions on Wireless Commmunications*, **4(2)**, 734–742, 2005.

A Proofs

A.1 Proof Of Theorem 1

Proof. Denote by $\Upsilon = \{\mathsf{f1}, \mathsf{f2}, \mathsf{f3}, \mathsf{f4}, \mathsf{f5}\}$ the set of keyed functions with the same secret key used in the UMTS/LTE protocol and $\Psi = \{\mathsf{r1}, \mathsf{r2}, \mathsf{r3}, \mathsf{r4}, \mathsf{r5}\}$ a set of random functions. The range of $\mathsf{f}j$ is equal to that of $\mathsf{r}j$, where $j \in \{1, 2, 3, 4, 5\}$. Define the sets $\mathsf{R} = \{\Upsilon, \mathsf{f8}, \mathsf{f9}\}$, $\mathsf{F} = \{\Psi, \mathsf{f8}, \mathsf{f9}\}$, $\mathsf{F}' = \{\Psi, \mathsf{r8}, \mathsf{f9}\}$ and $\mathsf{F}'' = \{\Psi, \mathsf{r8}, \mathsf{r9}\}$.

With these definitions, the experiment $\mathsf{Exp}^{\mathsf{s\text{-}anon}\text{-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R}, \mathsf{R}]$ is precisely the anonymity experiment of the UMTS/LTE protocol. The proof will proceed as follows; we will first switch the two usages of the set R to that of F, by a series of game hops, and then we will bound the probability of winning the experiment $\mathsf{Exp}^{\mathsf{s\text{-}anon}\text{-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}, \mathsf{F}]$. This latter task is accomplished by game hoping the two occurances of the set F to the set F' and then the set F''.

Switching from R to F: We first switch the experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}]$ into a modified experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{R}]$ such that the adversary $\mathcal{A}$ cannot obtain information about AK, CK and IK of the mobile station MS_{i_0}. The difference between the two experiments is as follows: In the modified experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{R}]$, instead of the outputs of Υ, we replay with the outputs of Ψ when $\mathcal{A}$'s query corresponds to MS_{i_0}. In the experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}]$, the function sets which the adversary $\mathcal{A}$ accesses are (R,R) for MS_{i_0} and MS_{i_1}, whereas in experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{R}]$ the functions sets the adversary $\mathcal{A}$ accesses are (F,R) for MS_{i_0} and MS_{i_1} respectively. Recall $\mathsf{R} = \{\Upsilon, \mathsf{f8}, \mathsf{f9}\}$ and $\mathsf{F} = \{\Psi, \mathsf{f8}, \mathsf{f9}\}$. Intuitively, the PRF-agility property of Υ should guarantee that this modification has only a negligible effect on the behavior of the adversary $\mathcal{A}$. More precisely we make the following claim.

Claim 1: We now claim that

$$|\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}]=1] - \Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{R}]=1]| = \mathsf{Adv}^{\mathsf{PR}}_{\Upsilon,\mathcal{B}}.$$

Proof of Claim 1: Specifically, we construct an adversary $\mathcal{B}$ against PRF-agility of the set Υ satisfying the above equality. Assume $\Upsilon = \{\mathsf{f1},\mathsf{f2},\mathsf{f3},\mathsf{f4},\mathsf{f5}\}$ is a set of keyed functions with the same secret key and $\Psi = \{\mathsf{r1},\mathsf{r2},\mathsf{r3},\mathsf{r4},\mathsf{r5}\}$ is a set of random functions. The range of $\mathsf{f}j$ is equal to that of $\mathsf{r}j$, where $j \in \{1,2,3,4,5\}$. The adversary $\mathcal{B}$ against the PRF-agility property has its own oracle Fn. On input (j,x), oracle Fn returns $\mathsf{f}j_K(x)$ or $\mathsf{r}j(x)$. The adversary $\mathcal{B}$ proceeds as in Figure 3. To simulate the experiment for $\mathcal{A}$, the adversary $\mathcal{B}$ generates $\mathsf{MS.SQN}_{i_0}$, $\mathsf{NET.SQN}_{i_0}$, START_{i_0} for MS_{i_0} and lets the key of its oracle K be the master key K_{i_0} of MS_{i_0}. The adversary $\mathcal{B}$ also generates K_{i_1}, $\mathsf{MS.SQN}_{i_1}$, $\mathsf{NET.SQN}_{i_1}$, START_{i_1} for MS_{i_1} and maintains the above parameters. Algorithm $\mathcal{B}$ then runs adversary $\mathcal{A}$.

Adversary $\mathcal{B}^b_{i_0,i_1}$
- $(\mathsf{K}_{i_1}, \mathbf{MS.SQN}, \mathbf{NET.SQN}, \mathbf{START}) \leftarrow \mathsf{Setup}()$
- $\mathsf{st} \leftarrow \mathcal{A}^{\mathsf{MS[X,R]}(\mathsf{id},x),\mathsf{NET[X,R]}(\mathsf{id},x)}$,where $\mathsf{X} = \{\mathsf{Fn},\mathsf{f8},\mathsf{f9}\}$
- $b \xleftarrow{\$} \{0,1\}$
- $\hat{b} \leftarrow \mathcal{A}^{\mathsf{MS[X,R]}(\mathsf{id},x),\mathsf{NET[X,F]}(\mathsf{id},x),\mathsf{Reveal}(i)}(\mathsf{st}, \mathsf{TMSI}_{i_b}, \mathsf{TMSI}_{i_{1-b}})$
- output 1 if $\hat{b} = b$, else output 0

Fig. 3. Adversary $\mathcal{B}$.

In the normal phase, when $\mathcal{A}$'s queries corresponds to MS_{i_1}, the set of functions $\mathcal{A}$ accesses is given by $\mathsf{R} = \{\Upsilon, \mathsf{f8}, \mathsf{f9}\}$. Specifically, if $\mathcal{A}$'s query corresponds to MS_{i_1}, $\mathcal{B}$ follows the processes of Figure 5 and Figure 6 and generates $(\mathsf{MAC},\mathsf{RES},\mathsf{CK},\mathsf{IK},\mathsf{AK})$ for MS_{i_1} by means of the set R. So $\mathsf{MAC} = \mathsf{f1}_{\mathsf{K}_{i_1}}(\mathsf{SQN}||\mathsf{RAND}||\mathsf{AMF})$, $\mathsf{XRES} = \mathsf{f2}_{\mathsf{K}_{i_1}}(\mathsf{RAND})$, $\mathsf{CK} = \mathsf{f3}_{\mathsf{K}_{i_1}}(\mathsf{RAND})$, $\mathsf{IK} = \mathsf{f4}_{\mathsf{K}_{i_1}}(\mathsf{RAND})$ and $\mathsf{AK} = \mathsf{f5}_{\mathsf{K}_{i_1}}(\mathsf{RAND})$. The algorithm $\mathcal{B}$ then computes $(\mathsf{KEYSTREAM}, \mathsf{MAC\text{-}I})$ by means of $(\mathsf{f8},\mathsf{f9})$ under $(\mathsf{CK},\mathsf{IK})$. When $\mathcal{A}$'s query corresponds to MS_{i_0}, the set of functions $\mathcal{A}$ accesses is $\mathsf{R} = \{\Upsilon,\mathsf{f8},\mathsf{f9}\}$ or $\mathsf{F} = \{\Psi,\mathsf{f8},\mathsf{f9}\}$ depending on $\mathcal{B}$'s oracle Fn. Specifically, $\mathcal{B}$ follows the processes of Figure 5 and Figure 6 but when $(\mathsf{MAC},\ \mathsf{RES},\ \mathsf{CK},\ \mathsf{IK},\mathsf{AK})$ are needed, $\mathcal{B}$ queries its oracle Fn to get back responses to answer $\mathcal{A}$. So $\mathcal{B}$ generates $(\mathsf{MAC},\mathsf{RES},\mathsf{CK},\mathsf{IK},\mathsf{AK})$ for MS_{i_0} by means of $\Upsilon = \{\mathsf{f1},\mathsf{f2},\mathsf{f3},\mathsf{f4},\mathsf{f5}\}$ under K or $\Psi = \{\mathsf{r1},\mathsf{r2},\mathsf{r3},\mathsf{r4},\mathsf{r5}\}$. In this case, $\mathsf{MAC} = \mathsf{f1}_K(\mathsf{SQN}||\mathsf{RAND}||\mathsf{AMF})$ or $\mathsf{MAC} = \mathsf{r1}(\mathsf{SQN}||\mathsf{RAND}||\mathsf{AMF})$; $\mathsf{XRES} = \mathsf{f2}_K(\mathsf{RAND})$ or $\mathsf{XRES} = \mathsf{r2}(\mathsf{RAND})$; $\mathsf{CK} = \mathsf{f3}_K(\mathsf{RAND})$ or $\mathsf{CK} = \mathsf{r3}(\mathsf{RAND})$; $\mathsf{IK} = \mathsf{f4}_K(\mathsf{RAND})$ or $\mathsf{IK} = \mathsf{r4}(\mathsf{RAND})$; $\mathsf{AK} = \mathsf{f5}_k(\mathsf{RAND})$ or $\mathsf{AK} = \mathsf{r5}(\mathsf{RAND})$. Then $\mathcal{B}$ generates $(\mathsf{KEYSTREAM},\ \mathsf{MAC\text{-}I})$ by means of $(\mathsf{f8},\mathsf{f9})$ under $(\mathsf{CK},\mathsf{IK})$. At the end of normal phase, $\mathcal{A}$ outputs some state information st and is given the two fresh TMSIs.

In the challenge phase, $\mathcal{B}$ picks $b \xleftarrow{\$} \{0,1\}$ and then answers $\mathcal{A}$'s query according to b. If $b = 1$, the view of $\mathcal{A}$ is exactly as in $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}]$. If $b = 0$ and Fn returns the outputs of Ψ, the view of $\mathcal{A}$ is exactly as

in $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{R}]$. Finally, $\mathcal{A}$ outputs a decision bit $\hat{b}$, and algorithm $\mathcal{B}$ outputs 1 if $\hat{b} = b$, else it outputs 0. So by definition 3 we have,

$$\begin{aligned}
&|\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}] = 1] - \Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{R}] = 1]| \\
&\quad = |\Pr[\Upsilon = \{\mathsf{f1},\mathsf{f2},\mathsf{f3},\mathsf{f4},\mathsf{f5}\} \xleftarrow{\$} \mathcal{F} : \mathcal{B}^{\Upsilon} = 1] \\
&\qquad - \Pr[\Psi = \{\mathsf{r1},\mathsf{r2},\mathsf{r3},\mathsf{r4},\mathsf{r5}\} \xleftarrow{\$} \Gamma : \mathcal{B}^{\Psi} = 1]| \\
&\quad = \mathsf{Adv}^{\mathsf{PR}}_{\Upsilon,\mathcal{B}}.
\end{aligned}$$

Thus proving Claim 1.

We then switch the experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A}}[\mathsf{F},\mathsf{R}]$ into a modified experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A}}[\mathsf{F},\mathsf{F}]$ such that the adversary $\mathcal{A}$ cannot obtain information about AK, CK and IK of the mobile station MS_{i_1}. The difference is that in $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A}}[\mathsf{F},\mathsf{F}]$, instead of the outputs of Υ, we replay with the outputs of Ψ when $\mathcal{A}$'s query corresponds to MS_{i_1}. In the experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A}}[\mathsf{F},\mathsf{F}](p)$, the functions $\mathcal{A}$ accesses are (F,F) whereas in the experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A}}[\mathsf{F},\mathsf{R}]$ the functions $\mathcal{A}$ accesses are (F,R). Recall $\mathsf{F} = \{\Upsilon,\mathsf{f8},\mathsf{f9}\}$ and $\mathsf{R} = \{\Psi,\mathsf{f8},\mathsf{f9}\}$, the PRF-agility property of Υ should guarantee that this modification has only a negligible effect on the behavior of the adversary $\mathcal{A}$.

Claim 2: We claim that

$$|\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{R}] = 1] - \Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}] = 1]| = \mathsf{Adv}^{\mathsf{PR}}_{\Upsilon,\mathcal{C}}.$$

Proof of Claim 2: This follows *Proof of Claim 1* similarly. The algorithm $\mathcal{C}$ against agility of Υ is similar to $\mathcal{B}$. The differences are as follows. The algorithm $\mathcal{C}$ generates the master key for MS_{i_0} and lets the key K of its oracle be the master key of MS_{i_1}. $\mathcal{C}$ then answers $\mathcal{A}$'s query by means of F when $\mathcal{A}$'s query corresponds to MS_{i_0} and answers by means of R or F (depending on the oracle Fn) when $\mathcal{A}$'s query corresponds to MS_{i_1}. We omit the details.

Putting Claim 1 and Claim 2 together we obtain:

$$\begin{aligned}
\mathsf{Adv}^{\text{s-anon}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}] &= |\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{R},\mathsf{R}] = 1] \\
&\quad - \Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{R}] = 1]| \\
&\quad + |\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{R}] = 1] \\
&\quad - \Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}] = 1]| \\
&\quad + \mathsf{Adv}^{\text{s-anon}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}] \\
&\leq \mathsf{Adv}^{\mathsf{PR}}_{\Upsilon,\mathcal{B}} + \mathsf{Adv}^{\mathsf{PR}}_{\Upsilon,\mathcal{C}} + \mathsf{Adv}^{\text{s-anon}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}]
\end{aligned}$$

Bounding $\mathsf{Adv}^{\text{s-anon}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}]$: Next we bound the advantage $\mathsf{Adv}^{\text{s-anon}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}]$ by the following modifications. First, we switch $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}]$ to a modified experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}',\mathsf{F}]$ for MS_{i_0} as follows. The difference is that in $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}',\mathsf{F}]$, if $\mathcal{A}$'s query corresponds to MS_{i_0}, then the ciphering keystream $\mathsf{KEYSTREAM}$ is computed from a random function $\mathsf{r8}$ rather than computed by means of $\mathsf{f8}$ under CK_{i_0}. The functions $\mathcal{A}$ accesses are F' and F for MS_{i_0} and MS_{i_1} respectively. Recall $\mathsf{F} = \{\Psi,\mathsf{f8},\mathsf{f9}\}$ and $\mathsf{F}' = \{\Psi,\mathsf{r8},\mathsf{f9}\}$. If $\mathsf{f8}$ is a pseudo random function, then this modification has only a negligible effect on the behavior of the adversary $\mathcal{A}$.

Claim 3: We claim that

$$|\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}]=1]-\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}',\mathsf{F}]=1]| \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f8},\mathcal{D}}.$$

Proof of Claim 3: Assume a prf-adversary $\mathcal{D}$ has its own oracle Fn. On input (x), oracle Fn returns $\mathsf{f8}_K(x)$ or $\mathsf{r8}(x)$, where r8 is a random function with the same range of f8. The adversary $\mathcal{D}$ proceeds as in Figure 4. To simulate the experiment for $\mathcal{A}$, adversary $\mathcal{D}$ first generates $\mathsf{MS.SQN}_{i_0}$, $\mathsf{MS.SQN}_{i_1}$, $\mathsf{NET.SQN}_{i_0}$, $\mathsf{NET.SQN}_{i_1}$, START_{i_0}, START_{i_1} for MS_{i_0} and MS_{i_1}. Then $\mathcal{D}$ runs $\mathcal{A}$. In the normal phase, if $\mathcal{A}$'s query corresponds to MS_{i_1} , then the functions $\mathcal{A}$ accesses are (F,F). If $\mathcal{A}$'s query corresponds to MS_{i_0}, then the functions $\mathcal{A}$ accesses are (F,F) or (F',F) depending on $\mathcal{B}$'s oracle Fn. In this experiment, $\mathcal{D}$ follows the processes of Figure 5 and Figure 6 and derives $(\mathsf{MAC},\mathsf{RES},\mathsf{CK},\mathsf{IK},\mathsf{AK})$ for MS_{i_0} and MS_{i_1} from random functions $\Psi=\{\mathsf{r1},\mathsf{r2},\mathsf{r3},\mathsf{r4},\mathsf{r5}\}$ first, i.e. $\mathsf{MAC}=\mathsf{r1}(\mathsf{SQN}||\mathsf{RAND}||\mathsf{AMF})$, $\mathsf{XRES}=\mathsf{r2}(\mathsf{RAND})$, $\mathsf{CK}=\mathsf{r3}(\mathsf{RAND})$, $\mathsf{IK}=\mathsf{r4}(\mathsf{RAND})$ and $\mathsf{AK}=\mathsf{r5}(\mathsf{RAND})$. If the cipher keystream is needed for MS_{i_1}, $\mathcal{D}$ computes the keystream by means of f8 under CK_{i_1} of MS_{i_1}. i.e., $\mathsf{KEYSTREAM}=\mathsf{f8}_{\mathsf{CK}_{i_1}}(\mathsf{COUNTER\text{-}C},\mathsf{BEARER},\mathsf{DIRECTION},\mathsf{LENGTH})$. If the cipher key stream is needed for MS_{i_0}, $\mathcal{D}$ queries its own oracle Fn to get back the output of f8 under K or r8,. i.e., either via $\mathsf{KEYSTREAM}=\mathsf{f8}_{\mathsf{CK}_{i_0}}(\mathsf{COUNTER\text{-}C},\mathsf{BEARER},\mathsf{DIRECTION},\mathsf{LENGTH})$ or via $\mathsf{KEYSTREAM}=\mathsf{r8}(\mathsf{COUNTER\text{-}C},\mathsf{BEARER},\mathsf{DIRECTION},\mathsf{LENGTH})$ depending on Fn. Note that Ψ is agile by definition, so both the cipher keys CK_{i_0} and CK_{i_1} are indistinguishable from random. At the end of normal phase, $\mathcal{A}$ outputs some state information st and is given two just allocated TMSIs.

Adversary $\mathcal{D}^b_{i_0,i_1}$
- $(\mathbf{MS.SQN},\mathbf{NET.SQN},\mathbf{START}) \leftarrow \mathsf{Setup}()$
- $\mathsf{st} \leftarrow \mathcal{A}^{\mathsf{MS[X,F]}(\mathsf{id},x),\mathsf{NET[X,F]}(\mathsf{id},x)}$,where $\mathsf{X}=\{\Psi,\mathsf{Fn},\mathsf{f9}\}$
- $b \xleftarrow{\$} \{0,1\}$
- $\hat{b} \leftarrow \mathcal{A}^{\mathsf{MS[X,F]}(\mathsf{id},x),\mathsf{NET[X,F]}(\mathsf{id},x)}(\mathsf{st},\mathsf{TMSI}_{i_b},\mathsf{TMSI}_{i_{1-b}})$
- output 1 if $\hat{b}=b$, else output 0

Fig. 4. Adversary $\mathcal{D}$.

In the challenge phase, $\mathcal{D}$ picks $b \xleftarrow{\$} \{0,1\}$ and then answers $\mathcal{A}$'s query according to b. In this case, if $b=1$, the view of $\mathcal{A}$ is just in $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}]$. If $b=0$ and Fn returns the outputs of r8, the view of $\mathcal{A}$ is just in $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}',\mathsf{F}]$. Finally, $\mathcal{A}$ outputs the decision bit $\hat{b}$, $\mathcal{D}$ outputs 1 if $\hat{b}=b$, else outputs 0. Therefore, $\mathcal{D}$ can determine which world it is in and break the prf of f8 if the followings hold: 1) the key K of the oracle Fn is equal to r3(RAND) 2) adversary $\mathcal{A}$ behaves differently when $\mathcal{D}$ is working with keyed function f8, as opposed to $\mathcal{D}$ working with random function r8. So by definition 2,

$$\begin{aligned}
&|\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}]=1]-\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}',\mathsf{F}]=1]|\\
&\quad=|\Pr[\mathsf{f8} \xleftarrow{\$} \mathcal{F}:\mathcal{B}^{\mathsf{f8}}=1]-\Pr[\mathsf{r8} \xleftarrow{\$} \Gamma_{\ell_1,\ell_2}:\mathcal{B}^{\mathsf{r8}}=1]|\\
&\quad=\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f8},\mathcal{C}}.
\end{aligned}$$

Thus proving Claim 3.

Now we switch $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}',\mathsf{F}]$ to a modified experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}]$ for MS_{i_0} as follow. The difference is that in $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}]$, if $\mathcal{A}$'s query corresponds to MS_{i_0}, then the message authentication tag is computed from a random function r9 rather than computed by means of f9 and IK_{i_0}. The functions $\mathcal{A}$ accesses are $(\mathsf{F}''',\mathsf{F}')$. Recall $\mathsf{F}''=\{\Psi,\mathsf{r8},\mathsf{f9}\}$ and $\mathsf{F}'''=\{\Psi,\mathsf{r8},\mathsf{r9}\}$. If f9 is a pseudo random function, then this modification has only a negligible effect on the behavior of the adversary $\mathcal{A}$.

Claim 4: We claim that

$$|\Pr[\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}',\mathsf{F}]=1]-\Pr[\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}]=1]|\leq \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f9},\mathcal{E}}.$$

Proof of Claim 4: The algorithm $\mathcal{E}$ against prf of $\mathsf{f9}$ is similar to $\mathcal{D}$. The differences are that $\mathcal{E}$ let the key of Fn be the key of $\mathsf{f9}$ when $\mathcal{A}$'s query corresponds to MS_{i_0}. $\mathcal{E}$ then answers by means of F' or F'' (depending on the oracle Fn) when $\mathcal{A}$'s query corresponds to MS_{i_0} and answers by means of F when $\mathcal{A}$'s query corresponds to MS_{i_1}. We omit the details.

We then switch $\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}]$ to a modified experiment $\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}']$ for MS_{i_1} as follow. The difference is that in $\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}']$, if $\mathcal{A}$'s query corresponds to MS_{i_1}, then the ciphering keystream $\mathsf{KEYSTREAM}$ is computed from a random function $\mathsf{r8}$ rather than computed by means of $\mathsf{f8}$ and CK_{i_1}. The functions $\mathcal{A}$ accesses are $(\mathsf{F}'',\mathsf{F}')$. Recall $\mathsf{F}=\{\Psi,\mathsf{f8},\mathsf{f9}\}$ and $\mathsf{F}'=\{\Psi,\mathsf{r8},\mathsf{f9}\}$. If $\mathsf{f8}$ is a pseudo random function, then this modification has only a negligible effect on the behavior of the adversary $\mathcal{A}$.

Claim 5: We claim that

$$|\Pr[\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}]=1]-\Pr[\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}']=1]|\leq \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f8},\mathcal{F}}.$$

Proof of Claim 5: This follows *Proof of Claim 3* similarly. The algorithm $\mathcal{F}$ against prf of $\mathsf{f8}$ is similar to $\mathcal{D}$. The differences are that $\mathcal{F}$ let the key of Fn be the key of $\mathsf{f8}$ when $\mathcal{A}$'s query corresponds to MS_{i_1}. $\mathcal{F}$ answers by means of F'' when $\mathcal{A}$'s query corresponds to MS_{i_0} and answers by means of F or F' (depending on the oracle Fn) when $\mathcal{A}$'s query corresponds to MS_{i_1}. We again omit the details.

Now we switch $\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}']$ to a modified experiment $\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}'']$ for MS_{i_1} as follow. The difference is that in $\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}'']$, if $\mathcal{A}$'s query corresponds to MS_{i_1}, then the message authentication tag is computed from a random function $\mathsf{r9}$ rather than computed by means of $\mathsf{f9}$ and IK_{i_1}. The functions $\mathcal{A}$ accesses are $[\mathsf{F}'',\mathsf{F}'']$. Recall $\mathsf{F}'=\{\Psi,\mathsf{r8},\mathsf{f9}\}$ and $\mathsf{F}''=\{\Psi,\mathsf{r8},\mathsf{r9}\}$. If $\mathsf{f9}$ is a pseudo random function, then this modification has only a negligible effect on the behavior of the adversary $\mathcal{A}$.

Claim 6: We claim that

$$|\Pr[\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}']=1]-\Pr[\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}'']=1]|\leq \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f9},\mathcal{G}}.$$

Proof of Claim 6: This follows *Proof of Claim 3* similarly. The algorithm $\mathcal{G}$ against prf of $\mathsf{f9}$ is similar to $\mathcal{D}$. The differences are that $\mathcal{G}$ let the key of Fn be the key of $\mathsf{f9}$ when $\mathcal{A}$'s query corresponds to MS_{i_1}. $\mathcal{G}$ answers by means of F'' when $\mathcal{A}$'s query corresponds to MS_{i_0} and answers by means of F' or F'' (depending on the oracle Fn) when $\mathcal{A}$'s query corresponds to MS_{i_1}. We again omit the details.

Claim 7: Now we claim that

$$\Pr[\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}0}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}'']=1]=\Pr[\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}1}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}'']=1]$$

Proof of Claim 7: The passive adversary $\mathcal{A}$ can break the anonymity of $\mathsf{Exp}^{\mathsf{s\text{-}anon\text{-}}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}'']$ only when one of the following cases occur:

- *Case 1:* $\mathcal{A}$ obtains information about the key or sequence number of the phone from the response of the oracle.
- *Case 2:* $\mathcal{A}$ queries same message in both normal and challenge phases for the same phone and the oracle returns the same response. For example, $\mathcal{A}$ queries message x with id_{i_0} in the normal phase and gets response y. Then $\mathcal{A}$ queries message x with TMSI_{i_b} in challenge phase. If $b=0$ and the oracle returns response y, $\mathcal{A}$ may link the two queries to the same phone MS_{i_0}.

Now we discuss the above two cases in turn. In the experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}'']$, no matter what $\mathcal{A}$ queries for MS_{i_0} or MS_{i_1}, the values $(\mathsf{MAC},\mathsf{RES},\mathsf{CK},\mathsf{IK},\mathsf{AK},\mathsf{KEYSTREAM},\mathsf{MAC\text{-}I})$ are all computed from random functions $\{\mathsf{r1},\mathsf{r2},\mathsf{r3},\mathsf{r4},\mathsf{r5},\mathsf{r8},\mathsf{r9}\}$ rather than from $\{\mathsf{f1},\mathsf{f2},\mathsf{f3},\mathsf{f4},\mathsf{f5},\mathsf{f8},\mathsf{f9}\}$ under K_{i_b} (in the original anonymity experiment). The key K_{i_0} of MS_{i_0} and the key K_{i_1} of MS_{i_1} are not used to generate anything. Moreover, the phone sequence number SQN is always masked by the random anonymity key $\mathsf{AK} = \mathsf{r5}(\mathsf{RAND})$ with a one-time pad, and SQN will be increased with each query. Thus the adversary $\mathcal{A}$ cannot obtain information about the sequence number from the public authentication token $\mathsf{AUTN} = \mathsf{SQN} \bigoplus \mathsf{AK}||\mathsf{AMF}||\mathsf{MAC}$. In other words, the hidden bit b does not effect this experiment from $\mathcal{A}$'s point of view. Therefore, the adversary $\mathcal{A}$ cannot get information from the response of the query, i.e. Case 1 cannot happen. Furthermore, since the random numbers, fresh numbers and counters used in functions $\{\mathsf{r1},\mathsf{r2},\mathsf{r3},\mathsf{r4},\mathsf{r5},\mathsf{r8},\mathsf{r9}\}$ are all different in each query, even if $\mathcal{A}$ queries the same message in different queries, the oracle always output different responses. The adversary $\mathcal{A}$ cannot distinguish the two phones by sending the same messages since the responses of the oracles are totally random from $\mathcal{A}$ point of view, i.e. Case 2 does not occur. So the adversary theb only strategy left to the adversary is to finally output the bit $\hat{b}$ by pure guessing. We therefore have that $\Pr[\mathsf{Exp}^{\text{s-anon-0}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}''] = 1] = 1/2$ and $\Pr[\mathsf{Exp}^{\text{s-anon-1}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}''] = 1] = 1/2$. Therefore, Claim 7 holds.

Now we bound $\mathsf{Adv}^{\text{s-anon}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}]$, by the probability differences between the above switches from $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}]$ to $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}'']$ via $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}',\mathsf{F}]$, $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}]$, and $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}']$. By Claim 3, 4, 5, 6, 7, we can bound

$$\begin{aligned}
\mathsf{Adv}^{\text{s-anon}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}] = {} & |\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F},\mathsf{F}] = 1] \\
& - |\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}',\mathsf{F}] = 1] \\
& + |\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}',\mathsf{F}] = 1] \\
& - |\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}] = 1] \\
& + |\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}] = 1] \\
& - |\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}'] = 1] \\
& + |\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}'] = 1] \\
& - |\Pr[\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}''] = 1] \\
& + \mathsf{Adv}^{\text{s-anon}}_{\Pi,\mathcal{A},i_0,i_1}[\mathsf{F}'',\mathsf{F}''] \\
\leq {} & \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f8},\mathcal{D}} + \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f9},\mathcal{E}} + \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f8},\mathcal{F}} + \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{f9},\mathcal{G}}.
\end{aligned}$$

From which the proof of the theorem follows.

A.2 Proof Of Theorem 2

Proof. Let $\mathcal{A}$ be an algorithm attacking anonymity for the dynamic case of the UMTS/LTE authentication and connection protocol. Denote by $\mathsf{Exp}^{\text{d-anon-}b}_{\Pi,\mathcal{A}}[\mathsf{R},\mathsf{R}]$ the experiment $\mathcal{A}$ performs for the dynamic case, $\mathsf{NET}^{\mathcal{A}}$ and $\mathsf{MS}^{\mathcal{A}}$ the oracles that $\mathcal{A}$ queries in $\mathsf{Exp}^{\text{d-anon-}b}_{\Pi,\mathcal{A}}[\mathsf{R},\mathsf{R}]$, and $\mathsf{Adv}^{\text{d-anon}}_{\Pi,\mathcal{A}}[\mathsf{R},\mathsf{R}]$ the advantage of $\mathcal{A}$.

Now we construct an adversary $\mathcal{B}$ attacking the anonymity for the static case by running algorithm $\mathcal{A}$ and simulating the required oracles needed by $\mathcal{A}$ in it's experiment. Denote $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{B},i_0,i_1}[\mathsf{R},\mathsf{R}]$ the experiment $\mathcal{B}$ performs for the static case, $\mathsf{NET}^{\mathcal{B}}$ and $\mathsf{MS}^{\mathcal{B}}$ the oracles that $\mathcal{B}$ queries in $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{B},i_0,i_1}[\mathsf{R},\mathsf{R}]$, and $\mathsf{Adv}^{\text{s-anon}}_{\Pi,\mathcal{B},i_0,i_1}[\mathsf{R},\mathsf{R}]$ the advantage of $\mathcal{B}$. We will show that

$$\mathsf{Adv}^{\text{d-anon}}_{\Pi,\mathcal{A}}[\mathsf{R},\mathsf{R}] \leq m(m-1) \cdot \mathsf{Adv}^{\text{s-anon}}_{\Pi,\mathcal{B},i_0,i_1}[\mathsf{R},\mathsf{R}],$$

where m is the number of phones registering to the network.

To simulate the environment for $\mathcal{A}$, the adversary $\mathcal{B}$ first generates master keys, sequence numbers, stat values for all phones except the two fixed phones MS_{i_0} and MS_{i_1} that $\mathcal{B}$ needs to distinguish. Then $\mathcal{B}$ runs $\mathcal{A}$ and answers $\mathcal{A}$'s queries to $\mathsf{NET}^{\mathcal{A}}$ and $\mathsf{MS}^{\mathcal{A}}$.

In the normal phase of the simulated experiment, if $\mathcal{A}$'s query corresponds to MS_i where $i \neq i_0$ and $i \neq i_1$, $\mathcal{B}$ follows the processes of Figure 5 and Figure 6 and answers $\mathcal{A}$'s query. Since $\mathcal{B}$ has master key of MS_i, $\mathcal{B}$ can answers the query perfectly. If $\mathcal{A}$'s query corresponds to MS_{i_0} and MS_{i_1}, $\mathcal{B}$ first queries its oracles $\mathsf{NET}^{\mathcal{B}}$ and $\mathsf{MS}^{\mathcal{B}}$ and then passes the responses to $\mathcal{A}$. At the end of normal phase, $\mathcal{A}$ outputs state information st and two identity indexes i'_0 and i'_1 of its choice. Then $\mathcal{B}$ checks whether (i_0, i_1) and (i'_0, i'_1) are equal. If $(i_0, i_1) \neq (i'_0, i'_1)$ than $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ distinguishes i_0 from i_1 by simulating this experiment for $\mathcal{A}$. $\mathcal{B}$ then simulates the challenge phase for $\mathcal{A}$. In the beginning of the challenge phase of the simulated experiment, $\mathcal{B}$ sends $\mathcal{A}$ the two TMSIs (TMSI_{i_b} and $\mathsf{TMSI}_{i_{1-b}}$) that $\mathcal{B}$ gets in the challenge phase of its own experiment $\mathsf{Exp}^{\text{s-anon-}b}_{\Pi,\mathcal{B},i_0,i_1}[\mathsf{R},\mathsf{R}]$. Then when $\mathcal{A}$ queries with the two TMSIs, $\mathcal{B}$ queries $\mathsf{NET}^{\mathcal{B}}$ and $\mathsf{MS}^{\mathcal{B}}$ and passes the responses to $\mathcal{A}$. At the end of the challenge phase, $\mathcal{A}$ halts with output $\hat{b}$, $\mathcal{B}$ then takes $\hat{b}$ as its output.

In the case of $(i_0, i_1) = (i'_0, i'_1)$, the bit $\hat{b}$ (that $\mathcal{A}$ outputs in $\mathcal{B}$'s simulation and $\mathcal{B}$ takes as its output) is with respect to the two phones MS_{i_0} and MS_{i_1} (that $\mathcal{B}$ needs to distinguish). So the adversary $\mathcal{B}$ wins (i.e. $\mathcal{B}$ breaks anonymity for static case) if the $\mathcal{B}$ does not abort and $\mathcal{A}$ wins (i.e. $\mathcal{A}$ breaks anonymity for dynamic case) . Therefore, we have

$$\Pr[\mathcal{B} \text{ wins}] = \Pr[\mathcal{B} \text{ does not abort} \wedge \mathcal{A} \text{ wins}].$$

Note the event that $\mathcal{B}$ does not abort and the event that $\mathcal{A}$ wins do not effect each other, the two events are independent, so

$$\Pr[\mathcal{B} \text{ does not abort} \wedge \mathcal{A} \text{ wins}] = \Pr[\mathcal{B} \text{ does not abort}] \cdot \Pr[\mathcal{A} \text{ wins}].$$

Since there are m phones registering to the network. The fixed indexes (i_0, i_1) to be distinguished for $\mathcal{B}$ need to match the selected indexes (i'_0, i'_1) of $\mathcal{A}$ (i.e. $i_0 = i'_0$ and $i_1 = i'_1$). This happens with probability at least $\frac{1}{m} \cdot \frac{1}{m-1}$, i.e.

$$\Pr[\mathcal{B} \text{ does not abort}] \geq \frac{1}{m(m-1)}.$$

We therefore derive

$$\begin{aligned}\Pr[\mathcal{B} \text{ wins}] &= \Pr[\mathcal{B} \text{ does not abort} \wedge \mathcal{A} \text{ wins}] \\ &= \Pr[\mathcal{B} \text{ does not abort}] \cdot \Pr[\mathcal{A} \text{ wins}] \\ &\geq \frac{1}{m(m-1)} \cdot \Pr[\mathcal{A} \text{ wins}].\end{aligned}$$

Finally we bound the advantage of $\mathcal{A}$ by

$$\mathsf{Adv}^{d\text{-anon}}_{\Pi,\mathcal{A}}[\mathsf{R},\mathsf{R}] \leq m(m-1) \cdot \mathsf{Adv}^{s\text{-anon}}_{\Pi,\mathcal{B},i_0,i_1}[\mathsf{R},\mathsf{R}].$$

B Figures

$\mathsf{net}[\{\{\mathsf{h1}, \mathsf{h2}, \mathsf{h3}, \mathsf{h4}, \mathsf{h5}\}, \mathsf{h8}, \mathsf{h9}\}](\mathsf{K}_i, \mathsf{NET.pc}_i, x)$

- if $\mathsf{NET.pc}_i = 1$ and $x =$(START$_i$, security capability) //receive START$_i$, supported integrity and cipher algorithms of the phone
  - store security capability of the phone
  - store START$_i$ in **START** according to index i
  - RAND $\xleftarrow{\$} \{0,1\}^{128}$
  - SQN $\leftarrow$ SQN.Gen(NET.SQN$_i$)
  - NET.SQN$_i \leftarrow$ NET.SQN$_i + 1$
  - MAC $\leftarrow \mathsf{h1}_{\mathsf{K}_i}$(SQN||RAND||AMF)
  - XRES $\leftarrow \mathsf{h2}_{\mathsf{K}_i}$(RAND)
  - CK$_i \leftarrow \mathsf{h3}_{\mathsf{K}_i}$(RAND)
  - IK$_i \leftarrow \mathsf{h4}_{\mathsf{K}_i}$(RAND)
  - AK$_i \leftarrow \mathsf{h5}_{\mathsf{K}_i}$(RAND)
  - AUTN $\leftarrow$ SQN $\bigoplus$ AK$_i$||AMF||MAC
  - NET.pc$_i \leftarrow 2$
  - return RAND||AUTN //*User Authentication Request*
- if $\mathsf{NET.pc}_i = 2$ and $x =$ RES //receive *User Authentication Response*
  - if RES $\neq$ XRES then abort
  - select integrity and encryption algorithms supported by the phone
  - FRESH $\leftarrow$ FRESH.Gen(k)
  - COUNTER-I $\leftarrow$ COUNTER-I.Gen(START$_i$)
  - START$_i \leftarrow$ START.Update(START$_i$, COUNTER-I)
  - m_S be the *Security Mode Command* message
  - MAC-I$_S \leftarrow \mathsf{h9}_{\mathsf{IK}_i}$(COUNTER-I, m_S, 1, FRESH)
  - NET.pc$_i \leftarrow 3$
  - return (m_S, FRESH, MAC-I$_S$) //*Security Mode Command*
- if $\mathsf{NET.pc}_i = 3$ and $x = (m_i, \mathsf{FRESH}, \mathsf{MAC\text{-}I}_i)||\mathsf{pc}$ //receive *Security Mode Complete*
  - COUNTER-I $\leftarrow$ COUNTER-I.Gen(START$_i$)
  - START$_i \leftarrow$ START.Update(START$_i$, COUNTER-I)
  - if MAC-I$_i \neq \mathsf{h9}_{\mathsf{IK}_i}$(COUNTER-I, m_i, 0, FRESH) then abort
  - if pc = 4, 6 or 7, NET.pc$_i \leftarrow$ pc else abort
  - return "OK"
- if $\mathsf{NET.pc}_i = 4$ and $x =$ allocate||pc //start TMSI *Allocation*
  - TMSI$_{i_n} \leftarrow$ TMSI.Gen(p)
  - COUNTER-C $\leftarrow$ COUNTER-C.Gen(START$_i$)
  - START$_i \leftarrow$ START.Update(START$_i$, COUNTER-C)
  - FRESH $\leftarrow$ FRESH.Gen(k)
  - KEYSTREAM $\leftarrow \mathsf{h8}_{\mathsf{CK}_i}$(COUNTER-C, BEARER, 1, $|m|$)
  - $c_{\mathsf{TMSI}} \leftarrow$ KEYSTREAM $\oplus$ TMSI$_{i_n}$
  - COUNTER-I $\leftarrow$ COUNTER-I.Gen(START$_i$)
  - START$_i \leftarrow$ START.Update(START$_i$, COUNTER-I)
  - MAC-I$_S \leftarrow \mathsf{h9}_{\mathsf{IK}_i}$(COUNTER-I, c_{TMSI}, 1, FRESH)
  - NET.pc$_i \leftarrow 5$
  - return (c_{TMSI}, FRESH, MAC-I$_S$) //TMSI *Allocation Command*
- if $\mathsf{NET.pc}_i = 5$ and $x = (\mathsf{ack}, \mathsf{MAC\text{-}I}_i)||\mathsf{pc}$ //receive TMSI *allocation complete*
  - if MAC-I$_i \neq \mathsf{h9}_{\mathsf{IK}_i}$(COUNTER-I, ack, 0, FRESH) then abort
  - if pc = 6 or 7 NET.pc$_i \leftarrow$ pc, else abort
  - return "OK"
- if $\mathsf{NET.pc}_i = 6$ and $x = m||\mathsf{pc}$ //start *Data Transmission*
  - COUNTER-C $\leftarrow$ COUNTER-C.Gen(START$_i$)
  - START$_i \leftarrow$ START.Update(START$_i$, COUNTER-C)
  - KEYSTREAM $\leftarrow \mathsf{h8}_{\mathsf{CK}_i}$(COUNTER-C, BEARER, 1, $|m|$)
  - $c_S \leftarrow$ KEYSTREAM $\oplus\, m$
  - COUNTER-I $\leftarrow$ COUNTER-C.Gen(START$_i$)
  - START$_i \leftarrow$ START.Update(START$_i$, COUNTER-I)
  - FRESH $\leftarrow$ FRESH.Gen(k)
  - MAC-I$_S \leftarrow \mathsf{h9}_{\mathsf{IK}_i}$(COUNTER-I, c_S, 1, FRESH)
  - if pc = 4 or 7, NET.pc$_i \leftarrow$ pc, else abort
  - return (c_S, FRESH, MAC-I$_S$)
- if $\mathsf{NET.pc}_i = 7$ and $x = (c_i, \mathsf{FRESH}, \mathsf{MAC\text{-}I}_i)||\mathsf{pc})$ //receive transmitted message
  - COUNTER-I $\leftarrow$ COUNTER-C.Gen(START$_i$)
  - START$_i \leftarrow$ START.Update(START$_i$, COUNTER-I)
  - if MAC-I $\neq \mathsf{h9}_{\mathsf{IK}}$(COUNTER-I, c_i, 0, FRESH) then abort
  - COUNTER-C $\leftarrow$ COUNTER-C.Gen(START$_i$)
  - START$_i \leftarrow$ START.Update(START$_i$, COUNTER-C)
  - KEYSTREAM $\leftarrow \mathsf{h8}_{\mathsf{CK}_i}$(COUNTER-C, BEARER, 0, $|m|$)
  - $m_i \leftarrow$ KEYSTREAM $\oplus\, c_i$
  - if pc = 4 or 6, NET.pc$_i \leftarrow$ pc, else abort
  - return "OK"
- else abort

Fig. 5. net function for NET oracle

$\mathsf{ms}[\{\{\mathsf{h1}, \mathsf{h2}, \mathsf{h3}, \mathsf{h4}, \mathsf{h5}\}, \mathsf{h8}, \mathsf{h9}\}](\mathsf{K}_i, \mathsf{MS.pc}_i, x)$

- if $\mathsf{MS.pc}_i = 1$ and $x = \mathsf{init}$ //start communication
  - $\mathsf{MS.pc}_i \leftarrow 2$
  - return START_i, security capability (supported integrity and cipher algorithms of the phone)
- if $\mathsf{MS.pc}_i = 2$ and $x = \mathsf{RAND}||\mathsf{AUTN}$ //receive *User Authentication Request*
  - parse x as $x_1||x_2||x_3||x_4$ where $x_1 = \mathsf{RAND}$, $x_2 = \mathsf{SQN} \bigoplus \mathsf{AK}$, $x_3 = \mathsf{AMF}$, $x_4 = \mathsf{MAC}$
  - $\mathsf{AK}_i \leftarrow \mathsf{h5}_{\mathsf{K}_i}(x_1)$
  - $\mathsf{SQN} \leftarrow x_2 \bigoplus \mathsf{AK}_i$
    - if $\mathsf{SQN} > \mathsf{MS.SQN}_i$ then $\mathsf{MS.SQN}_i \leftarrow \mathsf{SQN}$ else abort
  - if $x_4 = \mathsf{h1}_{\mathsf{K}_i}(\mathsf{SQN}||x_1||x_3)$ then $\mathsf{RES} \leftarrow \mathsf{h2}_{\mathsf{K}_i}(x_1)$ else abort
  - $\mathsf{CK}_i \leftarrow \mathsf{h3}_{\mathsf{K}_i}(x_1)$
  - $\mathsf{IK}_i \leftarrow \mathsf{h4}_{\mathsf{K}_i}(x_1)$
  - $\mathsf{MS.pc}_i \leftarrow 3$
  - return RES //*User Authentication Response*
- if $\mathsf{MS.pc}_i = 3$ and $x = (m_S, \mathsf{FRESH}, \mathsf{MAC\text{-}I}_S)||\mathsf{pc}$ //receive *Security Mode Command*
  - $\mathsf{COUNTER\text{-}I} \leftarrow \mathsf{COUNTER\text{-}I.Gen}(\mathsf{START}_i)$
  - $\mathsf{START}_i \leftarrow \mathsf{START.Update}(\mathsf{START}_i, \mathsf{COUNTER\text{-}I})$
  - if $\mathsf{MAC\text{-}I}_S \neq \mathsf{h9}_{\mathsf{IK}_i}(\mathsf{COUNTER\text{-}I}, m_S, 1, \mathsf{FRESH})$ then abort
  - control security capability
  - let m_i be *Security Mode Complete* message
  - $\mathsf{COUNTER\text{-}I} \leftarrow \mathsf{COUNTER\text{-}I.Gen}(\mathsf{START}_i)$
  - $\mathsf{START}_i \leftarrow \mathsf{START.Update}(\mathsf{START}_i, \mathsf{COUNTER\text{-}I})$
  - $\mathsf{MAC\text{-}I}_i \leftarrow \mathsf{h9}_{\mathsf{IK}_i}(\mathsf{COUNTER\text{-}I}, m_i, 0, \mathsf{FRESH})$
  - if $\mathsf{pc} = 4$, 5 or 6, $\mathsf{MS.pc}_i \leftarrow \mathsf{pc}$, else abort
  - return $(m_i, \mathsf{FRESH}, \mathsf{MAC\text{-}I}_i)$ //*Security Mode Complete*
- if $\mathsf{MS.pc}_i = 4$ and $x = (c_{\mathsf{TMSI}}, \mathsf{FRESH}, \mathsf{MAC\text{-}I}_S)||\mathsf{pc}$ //receive TMSI allocation command
  - $\mathsf{COUNTER\text{-}I} \leftarrow \mathsf{COUNTER\text{-}I.Gen}(\mathsf{START}_i)$
  - $\mathsf{START}_i \leftarrow \mathsf{START.Update}(\mathsf{START}_i, \mathsf{COUNTER\text{-}I})$
  - if $\mathsf{MAC\text{-}I} \neq \mathsf{h9}_{\mathsf{IK}}(\mathsf{COUNTER\text{-}I}, c_{\mathsf{TMSI}}, 1, \mathsf{FRESH})$ then abort
  - $\mathsf{COUNTER\text{-}C} \leftarrow \mathsf{COUNTER\text{-}C.Gen}(\mathsf{START}_i)$
  - $\mathsf{START}_i \leftarrow \mathsf{START.Update}(\mathsf{START}_i, \mathsf{COUNTER\text{-}C})$
  - $\mathsf{KEYSTREAM} \leftarrow \mathsf{h8}_{\mathsf{CK}_i}(\mathsf{COUNTER\text{-}C}, \mathsf{BEARER}, 1, |m|)$
  - $\mathsf{TMSI}_i \leftarrow \mathsf{KEYSTREAM} \oplus c_{\mathsf{TMSI}}$
  - $\mathbf{ID}_i \leftarrow \mathsf{TMSI}_{i_n}$
  - $\mathbf{Revealed}_i \leftarrow \mathsf{false}$
  - let ack be TMSI *Allocation Complete* acknowledgment
  - $\mathsf{MAC\text{-}I}_i \leftarrow \mathsf{h9}_{\mathsf{IK}_i}(\mathsf{COUNTER\text{-}I}, \mathsf{ack}, 0, \mathsf{FRESH})$
  - if $\mathsf{pc} = 5$ or 6, $\mathsf{MS.pc}_i \leftarrow \mathsf{pc}$
  - return $(\mathsf{ack}, \mathsf{MAC\text{-}I}_i)$ //TMSI *Allocation Complete*
- if $\mathsf{MS.pc}_i = 5$ and $x = m||\mathsf{pc}$ //start *Data Transmission*
  - $\mathsf{COUNTER\text{-}C} \leftarrow \mathsf{COUNTER\text{-}C.Gen}(\mathsf{START}_i)$
  - $\mathsf{START}_i \leftarrow \mathsf{START.Update}(\mathsf{START}_i, \mathsf{COUNTER\text{-}C})$
  - $\mathsf{KEYSTREAM} \leftarrow \mathsf{h8}_{\mathsf{CK}_i}(\mathsf{COUNTER\text{-}C}, \mathsf{BEARER}, 0, |m|)$
  - $c_i \leftarrow \mathsf{KEYSTREAM} \oplus m$
  - $\mathsf{COUNTER\text{-}I} \leftarrow \mathsf{COUNTER\text{-}I.Gen}(\mathsf{START}_i)$
  - $\mathsf{START}_i \leftarrow \mathsf{START.Update}(\mathsf{START}_i, \mathsf{COUNTER\text{-}I})$
  - $\mathsf{FRESH} \leftarrow \mathsf{FRESH.Gen}(k)$
  - $\mathsf{MAC\text{-}I}_i \leftarrow \mathsf{h9}_{\mathsf{IK}_i}(\mathsf{COUNTER\text{-}I}, c_i, 0, \mathsf{FRESH})$
  - if $\mathsf{pc} = 4$ or 6, $\mathsf{MS.pc}_i \leftarrow \mathsf{pc}$, else abort
  - return $(c_i, \mathsf{FRESH}, \mathsf{MAC\text{-}I}_i)$
- if $\mathsf{MS.pc}_i = 6$ and $x = (c_S, \mathsf{FRESH}, \mathsf{MAC\text{-}I}_i)||\mathsf{pc}$ //receive transmitted message
  - $\mathsf{COUNTER\text{-}I} \leftarrow \mathsf{COUNTER\text{-}I.Gen}(\mathsf{START}_i)$
  - $\mathsf{START}_i \leftarrow \mathsf{START.Update}(\mathsf{START}_i, \mathsf{COUNTER\text{-}I})$
  - if $\mathsf{MAC\text{-}I}_S \neq \mathsf{h9}_{\mathsf{IK}}(\mathsf{COUNTER\text{-}I}, c_S, 1, \mathsf{FRESH})$ then abort
  - $\mathsf{COUNTER\text{-}C} \leftarrow \mathsf{COUNTER\text{-}C.Gen}(\mathsf{START}_i)$
  - $\mathsf{START}_i \leftarrow \mathsf{START.Update}(\mathsf{START}_i, \mathsf{COUNTER\text{-}C})$
  - $\mathsf{KEYSTREAM} \leftarrow \mathsf{h8}_{\mathsf{CK}_i}(\mathsf{COUNTER\text{-}C}, \mathsf{BEARER}, 1, |m|)$
  - $m_S \leftarrow \mathsf{KEYSTREAM} \oplus c_S$
  - if $\mathsf{pc} = 4$ or 5, $\mathsf{MS.pc}_i \leftarrow \mathsf{pc}$, else abort
  - return "OK"
- else abort

Fig. 6. ms function for MS oracle